**Assignment -01 Answers:**

1. Local variables and function parameters are usually stored on the **thread's stack**. Some optimizers will put a few of them in CPU registers, but it's hard to control that. Global variables and static variables are stored in the process's static data or BSS segment.
2. Global variables are stored in **the data section.**
3. The resources assigned to a process- **CPU and memory**
4. Process identification is **a set of activities aiming to systematically define the set of business processes of a company and establish clear criteria for prioritizing them**. The output of process identification is a process architecture, which represents the business processes and their interrelations.
5. **CPU scheduler** selects a process among the processes that are ready to execute and allocates CPU to one of them.
6. The objective/principle which should be kept in view while selecting a scheduling processes are the following −

a. Fairness − All processes should be treated the same. No process should suffer indefinite postponement.

b. Maximize throughput − Attain maximum throughput. The largest possible number of processes per unit time should be serviced.

c. Predictability − A given job should run in about the same predictable amount of time and at about the same cost irrespective of the load on the system.

d. Maximum resource usage − The system resources should be kept busy. Indefinite postponement should be avoided by enforcing priorities.

e. Controlled Time − There should be control over the different times.

i. Response time

ii. Turnaround time

iii. Waiting time

1. 1.First-Come, First-Served (FCFS) Scheduling.

2.Shortest-Job-Next (SJN) Scheduling.

3.Priority Scheduling.

4.Shortest Remaining Time.

5.Round Robin (RR) Scheduling.

6.Multiple-Level Queues Scheduling.

1. **Single core** -A processor that has only one or single core is called Unicore Processor or Uniprocessor.

**Multi core** -A processor that has more than one core is called Multicore Processor.

1. More than one processing core to execute **two processes** in parallel.
2. Step 1: Fetch instruction. Execution cycle starts with fetching instruction from main memory. ...

Step 2: Decode instruction. ...

Step 3: Perform ALU operation. ...

Step 4: Access memory. ...

Step 5: Update Register File. ...

Step 6: Update the PC (Program Counter)

1. 1. Process ID. When a process is created, a unique id is assigned to the process which is used for unique identification of the process in the system.

2. Program counter. ...

3. Process State. ...

4. Priority. ...

5. General Purpose Registers. ...

6. List of open files. ...

7. List of open devices.

1. 1.New. This is the state when the process has just been created. ...

2.Ready. In the ready state, the process is waiting to be assigned the processor by the short term scheduler, so it can run. ...

3.Ready Suspended. ...

4.Running. ...

5.Blocked. ...

6.Blocked Suspended. ...

7.Terminated.

1. Single CPU systems use scheduling and can achieve multi-tasking because the time of the processor is time-shared by several processes so allowing each process to advance in parallel. So a process runs for some time and another waiting gets a turn
2. A context switch occurs **when a computer's CPU switches from one process or thread to a different process or thread**. Context switching allows for one CPU to handle numerous processes or threads without the need for additional processors.
3. Concurrency is the task of running and managing the multiple computations at the same time. While parallelism is the task of running multiple computations simultaneously.
4. Establishing priorities is necessary **in order to complete everything that needs to be done**. Prioritization is important because it with allow you to give your attention to tasks that are important and urgent so that you can later focus on lower priority tasks.
5. You can list running processes using the **ps command** (ps means process status). The ps command displays your currently running processes in real-time. This will display the process for the current shell with four columns: PID returns the unique process ID.
6. **Pstree command** in Linux that shows the running processes as a tree which is a more convenient way to display the processes hierarchy and makes the output more visually appealing. The root of the tree is either init or the process with the given pid.
7. **ps command** is used to list the currently running processes and their PIDs along with some other information depends on different options.
8. **Init process** is the mother (parent) of all processes on the system, it's the first program that is executed when the Linux system boots up; it manages all other processes on the system.
9. A new process can be created by the **fork () system call**. The new process consists of a copy of the address space of the original process. fork () creates new process from existing process. Existing process is called the parent process and the process is created newly is called child process.
10. The **process control block** is kept in a memory area that is protected from the normal user access. This is done because it contains important process information.
11. exit() terminates the calling process without executing the rest code which is after the exit() function.
12. **\_exit()** won't flushes the stdio buffer while **exit()** flushes the stdio buffer prior to exit.
13. \_exit() closes all open file descriptors and directory streams in the caller.
14. The exit() function shall then **flush all open streams with unwritten buffered data**, close all open streams, and remove all files created by tmpfile().
15. control+C is used to **abort the current task and regain user control**.
16. ctrl z is used to **pause the process**. It will not terminate your program, it will keep your program in background.
17. File descriptor is an int whereas a FILE \* is a file pointer. The main difference is that **the latter is buffered while the former is not**. A file pointer ( FILE\* ) typically contains more information about the stream such as current location, end of file marker, errors on the stream etc.
18. After the directory server has exceeded the file descriptor limit of **1024 per process**, any new process and worker threads will be blocked.
19. when you call **fopen() and fileno()** function to check the descriptor then you can get same FD number in 2 different processes because it returns the index of fdtable which is per-process.
20. You can get the exit status of the child **via the first argument of wait() , or the second argument of waitpid() , and then using the macros WIFEXITED and WEXITSTATUS with it**. waitpid() will block until the process with the supplied process ID exits.
21. **Zombie Process**: The parent process reads the exit status of the child process which reaps off the child process entry from the process table.
22. Parents pass on traits or characteristics, such as **eye colour and blood type**, to their children through their genes. Some health conditions and diseases can be passed on genetically too. Sometimes, one characteristic has many different forms.